

[0046] CLAIMS

What is claimed is:

1. A method for maintaining a module type definition table by a statically configured portion of an operating system kernel, comprising:
 - dynamically creating a module type definition; and
 - updating an external module type definition table to include the module type definition at the direction of the static operating system kernel.
2. The method of claim 1, wherein dynamically creating a module type definition includes receiving an operator generated dynamically loadable kernel module (“DLKM”) type identifier.
3. The method of claim 1, wherein dynamically creating a module type definition includes receiving a computer generated dynamically loadable kernel module (“DLKM”) type identifier.
4. The method of claim 1, wherein creating a module type definition includes receiving at least one support module identifier associated with control logic operative to conduct pre-registration support.
5. The method of claim 1, wherein creating a module type definition includes receiving at least one support module identifier associated with control logic operative to conduct a registration function.
6. The method of claim 1, wherein creating a module type definition includes receiving at least one support module identifier associated with control logic operative to conduct post-registration support.
7. The method of claim 1, wherein creating a module type definition includes receiving at least one support module identifier associated with control logic operative to conduct pre-loading support.

8. The method of claim 1, wherein creating a module type definition includes receiving at least one support module identifier associated with control logic operative to conduct post-loading support.
9. The method of claim 1, wherein creating a module type definition includes receiving at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with a support module.
10. The method of claim 1, wherein creating a module type definition includes receiving at least one symbol name, each symbol name being respectively associated with a support module.
11. A system for maintaining a module type definition table, comprising:
 - module type detection logic for detecting that a module is of an undefined module type;
 - module type identification logic for assigning a new module type associated with the module;
 - support module identification logic for identifying at least one support module associated with the module;
 - support module loading logic for loading the at least one identified support module; and
 - module type definition logic for externally storing data defining the module type.
12. The system of claim 11, wherein the module type identification logic includes logic for receiving an operator generated dynamically loadable kernel module (“DLKM”) type identifier.
13. The system of claim 11, wherein the module type identification logic includes logic for receiving a computer generated dynamically loadable kernel module (“DLKM”) type identifier.

14. The system of claim 11, wherein the support module identification logic includes logic for identifying a support module operative to conduct pre-registration support.
15. The system of claim 11, wherein the support module identification logic includes logic for identifying a support module operative to conduct a registration function.
16. The system of claim 11, wherein the support module identification logic includes logic for identifying a support module operative to conduct post-registration support.
17. The system of claim 11, wherein the support module identification logic includes logic for identifying a support module operative to conduct pre-loading support.
18. The system of claim 11, wherein the support module identification logic includes logic for identifying a support module operative to conduct post-loading support.
19. The system of claim 11, wherein the support module identification logic is operative to receive at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with each of the at least one support module.
20. The system of claim 11, wherein the support module identification logic is operative to receive at least one symbol name, each symbol name being respectively associated with each of the at least one support module.

21. A computer-readable storage medium encoded with processing instructions for maintaining a module type definition table by a static operating system kernel, comprising:

- instructions to identify a module type of a first module;
- instructions to determine that the module type of the first module is undefined;
- instructions to identify data defining the module type of the first module; and
- instructions to store the data defining the module type in a location external to the static operating system kernel.

22. A computer-readable storage medium of claim 21 wherein the data defining the module type comprises a pre-loading support module.

23. A computer-readable storage medium of claim 21 wherein the data defining the module type comprises a post-loading support module.

24. A computer-readable storage medium of claim 21 wherein the data defining the module type comprises a pre-registration support module.

25. A computer-readable storage medium of claim 21 wherein the data defining the module type comprises a post-registration support module.

26. A static operating kernel comprising:

- logic to receive a request to load a module,
- logic to identify a module type of the module,
- logic to reference an external module type definition table,
- logic to identify at least one support module associated with the module type in the external module type definition table, and
- logic to load the module based upon the module type and the at least one support module associated with the module type.

27. The static operating kernel of claim 26 further comprising:

logic to identifying at least one module type not previously defined in the external module type definition table,

logic to dynamically define the at least one module type,

logic to dynamically update the external module type definition table with the dynamically defined at least one module type.

28. The static operating kernel of claim 27, where the logic to dynamically define the at least one external module type includes receiving an operator identified module type.

29. The static operating kernel of claim 27, where the logic to dynamically define the at least one external module type includes receiving at least one identified support modules from an operator.

30. A static operating system kernel comprising:

logic to receive a request to load a module,

logic to identify a module type associated with the module,

logic to reference an external module type reference table,

logic to determine the module type is undefined in the external module type reference table, and

means to dynamically load the module.

31. The static operating system kernel of claim 30, wherein the means to dynamically load the module comprises:

means to dynamically define the module type that is undefined in the module type reference table,

means to dynamically update the external module type definition table to include the defined module type,

logic to load the module based upon the updated external module type definition table.

32. The static operating system kernel of claim 31, wherein the means to dynamically define the module type that is undefined in the module type reference table comprises logic to receive at least one operator generated module type.
33. The static operating system kernel of claim 31, wherein the means to dynamically define the module type that is undefined in the module type reference table comprises logic to receive at least one software generated module type.
34. The static operating system kernel of claim 31, wherein the means to dynamically define the module type that is undefined in the module type reference table comprises logic to identify at least one support module associated with the module type.